



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/706,625	11/06/2000	David Francis Bacon	YOR920000359US2	8377

7590 07/02/2003  
Daniel P Morris  
IBM Corporation  
Intellectual Property Law Dept  
P O Box 218  
Yorktown Heights, NY 10598

EXAMINER

ZHEN, LI B

ART UNIT	PAPER NUMBER
----------	--------------

2126

DATE MAILED: 07/02/2003

5

Please find below and/or attached an Office communication concerning this application or proceeding.

10

**Office Action Summary**

Application No.

09/706,625

Applicant(s)

BACON ET AL.

Examiner

Li B. Zhen

Art Unit

2126

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☐ Responsive to communication(s) filed on \_\_\_\_.
- 2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-16 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-16 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- 11) ☐ The proposed drawing correction filed on \_\_\_\_ is: a) ☐ approved b) ☐ disapproved by the Examiner.  
If approved, corrected drawings are required in reply to this Office action.
- 12) ☐ The oath or declaration is objected to by the Examiner.

**Priority under 35 U.S.C. §§ 119 and 120**

- 13) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a) ☐ All b) ☐ Some \* c) ☐ None of:  
1. ☐ Certified copies of the priority documents have been received.  
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_.  
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).  
\* See the attached detailed Office action for a list of the certified copies not received.
- 14) ☒ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).  
a) ☐ The translation of the foreign language provisional application has been received.
- 15) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892) 4) ☐ Interview Summary (PTO-413) Paper No(s). \_\_\_\_.
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948) 5) ☐ Notice of Informal Patent Application (PTO-152)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) \_\_\_\_ 6) ☐ Other: \_\_\_\_.

## **DETAILED ACTION**

### ***Specification***

1. The abstract of the disclosure is objected to because the abstract exceed 150 words in length. Correction is required. See MPEP § 608.01(b).
2. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

### ***Claim Rejections - 35 USC § 102***

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1, 15, and 16 are rejected under 35 U.S.C. 102(b) as being anticipated by "Persistent Java Objects: A Proposal" [hereinafter Malhotra].

As to claim 1, Malhotra teaches an object-based virtual machine environment [Java applications; p. 1, Motivation and Heaps], middleware applications [transactions; p. 2, Object Persistence; p. 4, Persistence Summary and Alternate Proposal], storage for storing objects for running the applications [Stores and Heaps, p. 1], a system heap [persistent non-garbage collected heaps] which is not garbage collected [in persistent non-garbage collected heaps, objects are deleted only if the delete method is invoked on them; p. 2, Object Deletion; p. 4, Persistence Summary and Alternate Proposal], a

Art Unit: 2126

middleware heap [persistent garbage collected heaps] which is garbage collected [in persistent garbage collected heaps, objects that are not reachable from a persistent root are eventually garbage collected; p. 2, Object Deletion; p. 4, Persistence Summary and Alternate Proposal], and a transient heap [transient heap] which is cleared in between successive applications [when the program ends, all objects created in transient heaps disappear; p. 2, Object Persistence; p. 4, Persistence Summary and Alternate Proposal].

As to claim 15, an object-based virtual machine environment [Java applications; p. 1, Motivation and Heaps], middleware applications [transactions; p. 2, Object Persistence; p. 4, Persistence Summary and Alternate Proposal], storage for storing objects for running the applications [Stores and Heaps, p. 1], a system heap [persistent non-garbage collected heaps], a middleware heap [persistent garbage collected heaps], a transient heap [transient heap], performing garbage collection on the middleware heap and the transient heap [in persistent garbage collected heaps, objects that are not reachable from a persistent root are eventually garbage collected; p. 2, Object Deletion; p. 4, Persistence Summary and Alternate Proposal], but not on the system heap [in persistent non-garbage collected heaps, objects are deleted only if the delete method is invoked on them; p. 2, Object Deletion; p. 4, Persistence Summary and Alternate Proposal], and clearing the transient heap in between successive applications [when the program ends, all objects created in transient heaps disappear; p. 2, Object Persistence; p. 4, Persistence Summary and Alternate Proposal].

As to claim 16, this is a product claim that corresponds to method claim 15; note the rejection to claim 15 above, which also meets this product claim.

***Claim Rejections - 35 USC § 103***

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 1 – 7, 15, and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over in view of U.S. Patent No. 6,275,985 to Ungar in view of “Concurrent Compacting Garbage Collection of a Persistent Heap” [hereinafter O’Toole].

As to claim 1, Ungar teaches an object-based virtual machine environment [virtual machine 116, Fig. 1; column 3, lines 60 – 67], storage for storing objects for running the applications [storage system 120, Fig. 1; column 3, lines 60 – 67], a system heap which is not garbage collected [system heap 312, Fig. 3; column 5, lines 1 – 45], and a transient heap [application heap 310, Fig. 3; column 5, lines 1 – 45]. Ungar does not appear to teach a middleware heap.

However, O’Toole teaches an object-based system with middleware [automatic storage management in transaction systems, object-oriented databases and persistent programming environment; p. 1, Abstract], a transient heap [Transitory Heaps for Temporary Data; p. 2 – 3, Design; p. 5, Section 4.1 The Transitory Heap], and a middleware heap [Persistent Heap; p. 6 – 7, Section 4.4 The Persistent Heap] which is garbage collected [garbage collection of the persistent heap; p. 7 – 8, Persistent GC].

It would have been obvious to a person of ordinary skill in the art at the time of the invention to apply the teaching of a middleware heap as taught by O'Toole to the invention of Ungar because this provides object persistence, and safe and efficient transactional storage [p. 1, see Abstract].

As to claim 15, Ungar teaches an object-based virtual machine environment [virtual machine 116, Fig. 1; column 3, lines 60 – 67], storage for storing objects for running the applications [storage system 120, Fig. 1; column 3, lines 60 – 67], a system heap [system heap 312, Fig. 3; column 5, lines 1 – 45], and a transient heap [application heap 310, Fig. 3; column 5, lines 1 – 45]. As to a middleware heap, see the rejection to claim 1 above.

As to claim 16, this is a product claim that corresponds to method claim 15; note the rejection to claim 15 above, which also meets this product claim.

As to claim 2, Ungar teaches the system classes for the virtual machine are loaded into the system heap, providing subsequent applications with the ability to use these classes without having to reload them [system heap 312 includes memory for use by the system that executes the application...execution of the system causes a number of objects to be allocated and manipulated in system heap 312; column 5, lines 25 – 30].

As to claim 3, Ungar as modified teaches reusable objects [persistent data] other than class objects that must persist between successive applications are stored in the middleware heap [promote all newly persistent data into volatile from-space; p. 7, Section 4.6 of O'Toole].

As to claims 4 and 5, Ungar as modified teaches, the middleware heap is garbage collected between successive applications [garbage collection of the persistent heap is initiated; p. 7, Persistent GC of O'Toole].

As to claim 6, Ungar as modified teaches the transient heap [transitory heap] is used for storing applications objects that are used for only the duration of the application [transitory heap contains only temporary objects that will be discarded when a failure occurs; p. 5, Section 4.1 of O'Toole].

As to claim 7, Ungar as modified teaches any objects in the transient heap that are eligible for use by the next application, and which are referenced by live objects in the system heap or middleware heap, are promoted to the middleware heap [update all of the transitory heap data to point to the newly promoted objects...this is done either by scanning the transitory heap form pointers to promoted objects; p. 7, Section 4.6 of O'Toole].

7. Claims 8 – 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ungar and O'Toole in view of U.S. Patent No. 6,249,793 to Printezis.

As to claim 8, Ungar as modified does not teach a card table.

However, Printezis teaches [column 6, lines 30 – 50] a card table [a data structure 70 referred to as a "card table"], in which each card corresponds to a portion of the middleware heap [card table 70 includes entries associated with memory regions within the heap 40...each entry in the card table 70 includes information regarding the memory region it is associated with], and the card is marked if the middleware heap potentially references an object in the transient heap [a card table entry may include a

Art Unit: 2126

dirty bit, which when set indicates that one or more pointers within the associated memory region have been modified since the dirty bit was last cleared].

It would have been obvious to a person of ordinary skill in the art at the time of the invention to apply the teaching of a card table as taught by Printezis to the invention of Ungar as modified because a card table entry may indicates that one or more pointers within the associated memory region have been modified or store information that includes time stamps reflecting a time at which a pointer in the associated region was last modified [column 6, lines 40 – 50 of Printezis].

As to claim 9, Ungar as modified teaches each card corresponds to a memory region having a size greater than the minimum size for an object [card table 70 includes entries associated with memory regions within the heap 40...each entry in the card table 70 includes information regarding the memory region it is associated with; column 6, lines 30 – 50 of Printezis].

As to 10, Ungar as modified teaches a card is marked whenever an object in the corresponding memory region is updated [a card table entry may include a dirty bit, which when set indicates that one or more pointers within the associated memory region have been modified since the dirty bit was last cleared; column 6, lines 30 – 50 of Printezis].

8. Claims 11 – 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ungar and O'Toole in view of U.S. Patent No. 5,950,008 to van Hoff.

As to claim 11, Ungar and O'Toole does not teach a middleware class loader and an application class loader.



However, van Hoff teaches a generating application specific class loaders [column 5, lines 23 – 38].

It would have been obvious to a person of ordinary skill in the art at the time of the invention to apply the teaching of generating application specific class loaders as taught by van Hoff to the invention of Ungar as modified because each application specific class loader contains information such as location information, fetch method for fetching object classes, linking method and load method that are specific to each class [column 5, lines 30 – 38 of van Hoff]. Obviously, a middleware specific class loader can be created to load middleware class objects.

As to claim 12, Ungar as modified teaches objects from classes loaded by the one or more system class loaders are created in the middleware heap or the transient heap depending on the current context [application specific class loader 300 itself contains both a class reference 302 and a data reference 304 to a data array 305; column 5, lines 23 – 40 of van Hoff].

As to claim 13, Ungar as modified teaches the current context is middleware if the method being run derives from a class loaded by the middleware class loader, and application if the method being run derives from a class loaded by the application class loader [application specific class loaders do not include an application specific class loader generator method 312, because it is assumed that all symbol references to be resolved by the application specific class loader will be successfully resolved by locating an associated object class on either the server identified by the location information 303; column 6, lines 1 – 10 of van Hoff].

As to claim 14, Ungar as modified teaches if the method being run derives from a class loaded by the one or more system class loaders, then the current context retains the value it had immediately before the method was run [a new application specific class loader is generated for that called method by the generator method...the new application specific class loader is then used to load the object class on that other server as well as to locate and load the object classes for any method calls made by that method; column 6, lines 10 – 41 of van Hoff].

***Conclusion***

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

"Defining and Handling Transient Fields in PJama" by Printezis et al. teaches transient and persistent heaps in a virtual machine.

"Transactions for Java" by Garthwaite et al. teaches transactions and persistence for Java.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Li B. Zhen whose telephone number is (703) 305-3406. The examiner can normally be reached on Mon - Fri, 8am - 4:30pm.

The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7239 for regular communications and (703) 746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Li B. Zhen  
Examiner  
Art Unit 2126



lbz  
June 25, 2003